Revision History for //depot/main/Applications/calendar/CalSequence.cpp

View | Sync | Diff Revs | Diff vs ClientFile | Describe Changelist | Help

Revision	File Name	Changelist	Date	Description
2	//depot/m...	6218	2000/01/27	updated to use globally monotonically increasing sequence numbers
1	//depot/m...	6181	2000/01/27	code to control host and device sequence numbers.

Actions

CalSequence.cpp

```
       #include "storage.h"
       #include "CalSequence.h"
       #include "caldb.h"

5      SequenceNumbers g_minSequenceNumbers;

       class iCalBookmarkFile: public PersistentStore
       {
       public:

10       iCalBookmarkFile()
         : PersistentStore("Calendar Sync Bookmark", &g_minSequenceNumbers, sizeof(SequenceNumbers))
         {
             load_data();

15       }

         virtual bool backup_restore_completed()
         {
             load_data();

20           return true;

         }

       };

       iCalBookmarkFile iCalBookmark;

25     void
       FindLastDeviceSequence()
       {
           // find the largest device sequence number
           unsigned int nMax = 0;

30         DbRecordHandle dbHandle;

           if ((dbHandle = Calendar::calendarDatabase.get_first_record()) != DB_NULL_HANDLE)
           {
35             do
               {
                   Calendar::CalendarRecord cr(dbHandle);
                   unsigned int nTest = cr.GetDeviceSequenceID();
                   if (nTest > nMax)
40                     nMax = nTest;
               } while ((dbHandle = Calendar::calendarDatabase.get_next_record(dbHandle)) != DB_NULL_HANDLE);

           }
```

CalSequence.cpp

```cpp
45          if (!nMax)
                nMax++;

50      if (nMax > g_minSequenceNumbers.lastDeviceSequence)
        {
                g_minSequenceNumbers.lastDeviceSequence = nMax;
                iCalBookmark.save_data();
        }
    }

55  unsigned int
    GetNextDeviceSequence()
    {
        // Disaster recovery.  If we can't find a last sequence number, go through the DB and guess.
60      if (!g_minSequenceNumbers.lastDeviceSequence)
                FindLastDeviceSequence();

        unsigned int nNext = ++g_minSequenceNumbers.lastDeviceSequence;

65      iCalBookmark.save_data();
        return nNext;
    }

    unsigned int
70  GetMinHostSequence()
    {
        return (g_minSequenceNumbers.minHostSequence);
    }

    void
75  SetLastHostSequence(unsigned int seq)
    {
        if (seq > g_minSequenceNumbers.lastHostSequence)
        {
                g_minSequenceNumbers.lastHostSequence = seq;
80              iCalBookmark.save_data();
        }
    }

    void
85  SetLastDeviceSequence(unsigned int seq)
    {
```